Electronic Version
Stylesheet Version v1.1.1

Description

Geometry-Based Symmetric Cryptosystem Method

Cross Reference To Related Applications

[0001] US Patent 5740250, April 1998, by Moh; US Patent 6038317, March 2000, by Magliveras et al; US Patent 6298137, October 2001, by Hoffstein et al; U.S. Provisional Patent Application No. 60/319,710, filed November 2002, by Berenstein and Chernyak.

Copyright Statement

[0002] This application claims priority from U.S. Provisional Patent Application No. 60/319,710, filed Nov. 19, 2002, and said Provisional Patent Application is incorporated herein by reference.

Background of Invention

[0003] Secure exchange of data between two parties, for example, between two computers, requires encryption. There are two general methods of encryption in use today, pri-

vate key encryption and public key encryption. A public key cryptosystem is one in which each party can publish their encryption process without compromising the security of the decryption process. The encryption process is popularly called a "trap-door" function. The public key cryptosystems are typically used for transmitting small amounts of data, such as credit card numbers, and they are also used to transmit a private key which is then used for private key encryption. Public key cryptosystems are generally slower than private key cryptosystems. Most of known public key cryptosystems have been recently broken using high computational power. In private key encryption, the two parties privately exchange the keys to be used for encryption and decryption. A widely used example of a private key cryptosystem is DES, the Data Encryption Standard. Such systems can be fast and secure, but they suffer the disadvantage that the two parties must exchange their keys privately. This problem is currently addressed by using of public key cryptosystems for private key distribution/sharing. The most famous key sharing method currently used is Diffie-Hellman protocol. However, in the situation when the same private key is used very frequently, especially in the case of large communi-

cation networks of trusted participants, the private key is vulnerable to attacks. Therefore, there is a necessity of the periodic change of the private keys. This later disadvantage amplifies the former disadvantage of the systems due to the necessity of synchronizing private keys among the participants of the communication network and thus may cause serious inconvenience for the participants. Most users, therefore, would find it desirable to have a cryptosystem which combines advantages of the private and public ones: relatively short, easily created keys with relatively high speed encryption and decryption processes, secure generation and/or distribution of private keys. In other words, the desirable solution has to be a synthesis of public and private cryptosystems.

[0004] It is among the objects of the invention to provide a cryptosystem with elements of public and private cryptosystems. In this cryptosystem both the encryption and decryption keys are composed out of non-secret outer component and a secret inner components in such a way that both components of the keys are relatively short and easily generated, and the encryption and decryption processes can be performed extremely rapidly.

[0005] It is also among the objects hereof to provide a cryptosys-

tem which has very low memory requirements and which depends on a variety of internal parameters that permit substantial flexibility in balancing security level, key length, encryption and decryption speed, memory requirements, and bandwidth. It is also among the objects of the invention to provide the cryptosystem capability for generating encryption/decryption transformations based both on the outer components of the keys and on cryptosystem's internal parameters so that knowledge of the outer components of the keys does not provide a slightest possibility for reconstruction of the inner components of the keys.

SUMMARY OF INVENTION

[0006] The symmetric encryption system of the present invention has short and easily created encryption/decryption keys and wherein the encryption and decryption processes are performed extremely rapidly, and has very low computer memory requirements. The encryption and decryption processes use the operations of addition and dot product of vectors in vector spaces over the field of real numbers or, more generally, over any ring. The cryptosystem of the present invention constructs encryption/decryption keys on the fly out of a chosen set of vectors of a given vector

space or, more generally, of a module over a given ring. Total length of the chosen vectors is comparable to or much shorter than the key lengths of the most widely used prior art cryptosystems. The present invention, while requiring extremely little computer memory (about 128 bits for the inner component of the encryption/decryption key), features an extremely high security level (at least 2^{178}), with encryption and decryption processes ranging from approximately two to three orders of magnitude faster than the prior art. Each encryption/decryption key of the cryptosystem hereof consists of an outer component and an inner component. The role of the outer component is played by a set of discrete data that, typically, is a finite sequence of positive integers. The role of the inner component (which also further referred to as "internal parameters") is played by continuous data. In one embodiment the internal parameters include a set of vectors of a given vector space. In another embodiment these parameters include, besides a set of vectors of a given vector space, a set of polynomial or rational automorphisms of this vector space. The encryption and decryption techniques are mutually symmetric and require the same time, amount of memory, and computational power. Therefore,

the same device can work both as the encryption and the decryption device. Only the outer component of the key determines in which mode, i.e., encryption or decryption, the device is currently working. Namely, the outer component of the key used for encryption a message can be transmitted along with the encrypted message so that the receiving device uses this public component as the public component of the decryption key. The present invention allows the internal parameters be chosen essentially at random from a large set of vectors. If the cryptosystem has m internal parameters each of which is a vector in the n-dimensional vector space V over the field of real numbers and the total size of the internal parameters is l binary bits, the security level is at least

[0007] $2^l \cdot (l - 1)!/[(n \cdot m - 1)!(l - n \cdot m)!$

[0008] (Actually the security level is much higher because the size l can be arbitrarily big and not public.) For example, if there are 4 private internal parameters that occupy 128 bits and belong to the 3-dimensional real vector space, the security level of the cryptosystem is at least $2^{128} \cdot 2^{50} = 2^{178}$.

[0009] The creation of an encryption transformation (from the space of plaintexts to the space of ciphertexts) requires a

choice of both an outer component and an inner component. Because of this the decryption transformation (from the space of ciphertexts to the space of plaintexts) cannot be reconstructed based solely on the outer component. Moreover, the continuous nature of the inner component leaves no chance to reconstruct it even in the case when both the outer component of the key and the ciphertext are publicly available. Even if, in addition to the outer component and the ciphertext, the plaintext is also publicly available, it is still impossible to reconstruct the inner component.

[0010] The outer components of keys of the cryptosystem of the present invention serve as the generators of both the encryption and decryption keys. In particular, the cryptosystem proposed by the present invention does not require the recipient of messages to communicate the outer component of the encryption key to the sender. In one embodiment, this outer component may be generated solely by the sender and sent to the recipient along with the encrypted message. In one embodiment, the outer component of the key can be attached as an initial segment of the transmitted message. In another embodiment, this outer component may be embedded in the encrypted

message at equal distances between the digits of the message.

[0011] An important feature of the cryptosystem hereof is a dynamic and highly secure update of encryption and decryption keys. The security of the keys is guaranteed by the fact that their update proceeds without exchange of the new keys between communicating parties. Instead of such an exchange, the outer component of the encryption key, as embedded into the transmitted message, determines a new decryption key, which, in its turn, triggers the generation of a new decryption transformation. This update does not require any change in the inner component. Actually, any transmitted message may trigger a new decryption key generation. Therefore, the cryptosystem of the present invention overcomes a serious disadvantage of major private key cryptosystems: in such private key cryptosystems as DES or AES the encryption key does not change over a certain period of time, which fact encourages attacks against the cryptosystem. Unlike this, each time as the outer component is changed the cryptosystem hereof generates a new encryption transformation.

[0012] In one embodiment the outer component of the key is a sequence of positive integers. This sequence may be gen-

erated at random by using any distribution of the first m natural numbers. The security of the symmetric cryptosystem of the present invention comes from the built-in geometric continuity of plaintexts and ciphertexts as points of vector spaces as well as from the continuity of the inner components of encryption/decryption keys. In other words, security of the proposed cryptosystem is guaranteed by the obvious mathematical fact that there are potentially uncountably many geometric transformations of a given vector space.

[0013] An embodiment of the invention is in the form of a method for encryption and decryption a digital message M, comprising the following steps: producing a module V over a ring R; producing an outer component P of the encryption key that includes sequence $(p_1, p_2, \ldots, p_k)$ where each member p_j of the sequence belongs to the set $\{1, 2, \ldots, m\}$ (the length k of the sequence is arbitrary and thus repetitions are allowed in the sequence); producing an inner component Q of the encryption key that includes elements $v_1, v_2, \ldots, v_m$ of V and automorphisms $g_1, g_2, \ldots, g_m$ of V; producing the encryption key K = (P; Q), where P is the outer component and Q is the inner component; producing an encryption automorphism T_e of V based on the

encryption key K, where T_e includes a composition of certain automorphisms T_1, T_2, ... , T_m of the module V which composition is performed in the order prescribed by P; producing an encrypted message element E as a function of a message element M in V and of the encryption automorphism T_e; transmitting the encrypted message element E along with the outer component P from one user to another; producing the outer component P' of the decryption key that includes sequence (p_k, p_{k-1}, ... , p_1), i.e., the sequence reversed of that involved in producing the outer component P of the encryption key; producing the decryption key K' = (P'; Q'), where P' is the outer component of the decryption key and Q' is the inner component of the decryption key which is equal to the inner component Q of the encryption key; producing a decryption automorphism T_d of V based on the decryption key K', where T_d includes a composition of the automorphisms T_1, T_2, ... , T_m , which composition is performed in the order prescribed by P', e.g., T_d is the inverse automorphism of T_e; determining the message element M as a function of the encrypted message element E and of the decryption automorphism T_d, where the function is the same as that one used in generation of E (that is, the decryption method is

symmetric to encryption: the decryption proceeds as the encryption, but with replacement of the outer component P with the outer component P').

[0014] Further features and advantages of the invention will become more readily apparent from the following detailed description when taken in conjunction with the accompanying drawings.

Brief Description of Drawings

[0015] FIG. 1 is a block diagram of a system that can be used in practicing embodiments of the invention.

[0016] FIG. 2 is a flow diagram of a symmetric encryption system which, when taken with the subsidiary flow diagrams referred to therein, can be used in implementing embodiments of the invention.

[0017] FIG. 3 is a flow diagram of a routine, in accordance with an embodiment of the invention, for generating outer component of the encryption key.

[0018] FIG. 4 is a flow diagram of a routine, in accordance with an embodiment of the invention, for generating the inner component of the encryption key using the outer component.

[0019] FIG. 5 is a flow diagram in accordance with an embodiment of the invention, for encryption a message using the

inner component of the encryption key.

[0020] FIG. 6 is a flow diagram of a routine, in accordance with an embodiment of the invention, for generating the inner component of the decryption key using the outer component.

[0021] FIG. 7 is a flow diagram in accordance with an embodiment of the invention, for decryption a message using the inner component of the encryption key.

[0022] FIG. 8 is a flow diagram of a routine, in accordance with another embodiment of the invention, for generating the inner component of the encryption key using the outer component.

[0023] FIG. 9 is a flow diagram in accordance with another embodiment of the invention, for generating the inner component of the decryption key using the outer component.

Detailed Description

[0024] FIG. 1 is a block diagram of a system that can be used in practicing embodiments of the invention. Two processor-based subsystems 101 and 151 are shown as being in communication over an insecure channel 100, which may be, for example, any wired or wireless communication channel such as a telephone or internet communication channel. The subsystem 101 includes processor 102 and

the subsystem 151 includes processor 152. When programmed in the manner to be described, the processors 102 and 152 and their associated circuits can be used to implement an embodiment of the invention and to practice an embodiment of the method of the invention. The processors 102 and 152 may each be any suitable processor, for example an electronic digital processor or microprocessor. It will be understood that any general purpose or special purpose processor, or other machine or circuitry that can perform the functions described herein, electronically, optically, or by other means, can be utilized. The processors may be, for example, Intel Pentium processors. The subsystem 101 will typically include memories 103, clock and timing circuitry 104, input/output functions 105 and monitor 106, which may all be of conventional types. Inputs can include a keyboard input as represented at 107. Communication is via transceiver 108, which may comprise a modem or any suitable device for communicating signals. The subsystem 151 in this illustrative embodiment can have a similar configuration to that of subsystem 101. The processor 152 has associated input/output circuitry 155, memories 153, clock and timing circuitry 154, and a monitor 156. Inputs include a

keyboard 157. Communication of subsystem 151 with the outside world is via transceiver 158 which, again, may comprise a modem or any suitable device for communicating signals.

[0025] The encryption and decryption techniques of an embodiment of the symmetric cryptosystem hereof use a cryptosystem based on an action of the infinite group on a vector space. The security of the symmetric cryptosystem of the present invention hereof comes from the built-in geometric continuity of plaintexts and ciphertexts as points of vector spaces as well as from the continuity of the inner component of encryption/decryption keys performing transformations between plaintexts and ciphertexts. In other words, security of the proposed cryptosystem is guaranteed by the obvious mathematical fact that there are potentially uncountably many geometric transformations of a given vector space.

[0026] The cryptosystem hereof is essentially a private key symmetric cryptosystem because both decryption and encryption keys are of the similar structure and are not publicly available. Another similarity is that in the cryptosystem hereof formation of both encryption and decryption keys depends on fixed secret internal parameters. However,

unlike in major private key symmetric cryptosystems like DES or AES there are in the cryptosystem hereof many different encryption/decryption keys corresponding to a chosen set of secret parameters. Namely, generation of a particular encryption/decryption key in the cryptosystem of the present invention depends, besides the fixed secret parameters, on a choice of certain publicly available data, which data is referred to as outer component. Another difference between the cryptosystem of the present invention and major private key cryptosystems is that the cryptosystem hereof requires neither sharing nor storing of encryption and decryption keys. In the cryptosystem hereof each message can be encrypted by its own encryption key independently of other messages. Each decryption key can be created upon receiving an encrypted message and does not have to be stored after the message has been decrypted. Thus the dynamic generation of encryption and decryption keys in the present invention eliminates the disadvantage of the major private key cryptosystems (like DES or AES) caused by the necessity of periodic change of the keys. Moreover, the present invention turns this disadvantage into a most efficient and attractive feature of the proposed cryptosystem. After a set of secret

internal parameters has been chosen, the encryption key depends entirely on the publicly available data, i.e., the outer component. However, this encryption key is not public itself and the publicly available data do not necessarily come from the potential recipient of the message. Moreover, the decryption key of the present invention does not have to be an exclusive property of the potential recipient of the message. Knowledge of the outer component does not allow for constructing an encryption key unless the secret internal parameters of the cryptosystem are available. Thus, construction or reconstruction of any key in the cryptosystem hereof requires both a set of secret internal parameters and an outer component. The same outer component is used for constructing both encryption and decryption keys.

[0027] So far there is no literature describing cryptosystem embodying a geometric principle underlying the system hereof. Apparently an approach that is the closest to the present invention is developed in U.S. Pat. No. 5,740,250 entitledTAME AUTOMORPHISMPUBLIC KEY SYSTEM by Moh. The idea of using polynomial automorphisms in cryptography was developed in the patent. However, this is perhaps the only similarity because the Moh"s patent ad-

dresses only the public key cryptosystem.

[0028] An embodiment of the cryptosystem hereof deals with the n-dimensional vector space V over the field of real numbers and a bilinear form L on V. A vector x in V can be written as an n-tuple of real numbers: $x = [x_1, x_2, \ldots, x_n]$. A bilinear form can be written as

[0029] $$L(x, y) = \Sigma l_{i,j} \cdot x_i \cdot y_j ,$$

[0030] where the summation is over all pairs (i,j) such that $1 \le i,j \le n$, and all $l_{i,j}$ are real numbers. The embodiment of the cryptosystem hereof depends on discrete parameters n and m, which are positive integers, and the set of continuous parameters: any vectors $v_1, v_2, \ldots, v_m$ of V. In an embodiment the coordinates of the vectors of the cryptosystem hereof are presented by decimal real numbers having totally l decimal digits (therefore, the average number of digits in each coordinate is $l/(n \cdot m)$). Therefore, the security level of the cryptosystem hereof is measured as the number of all such sets of parameters, i.e.,

[0031] $$10^l \cdot (l - 1)!/[(n \cdot m - 1)!(l - n \cdot m)!].$$

[0032] For example, if $n = 3$, $m = 4$, $l = 72$, the security level is measured as

[0033] $$10^{72} \cdot (72 - 1)!/[(3 \cdot 4 - 1)!\ (72 - 3 \cdot 4)!] \approx 2.5 \cdot 10^{84}$$

[0034] (Actually the security level is much higher because the total number l of the digits can be arbitrarily big and is not public.) The following is an example of an embodiment in accordance with the invention of a symmetric key cryptosystem. The small numbers $n = 3$, $m = 4$, $l \leq 24$ are used for ease of illustration, however, even with these small numbers the cryptosystem hereof is still cryptographically secure. Its security level is measured as at least $1.3 \cdot 10^{30} \approx 2^{100}$. In creating a symmetric cryptosystem in accordance with an embodiment hereof (and with the previously indicated small numbers for ease of illustration), a first step is to choose integer parameters m, n. Take, for example $n = 3$, $m = 4$. Next, the bilinear form L is chosen to be the standard Euclidean dot product on $V = R^3$, that is,

[0035] $L(x, y) = x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3$

[0036] for all x and y in R^3. Some sequence of vectors v_1, v_2, v_3, v_4 is chosen as follows: v_1=[1,21,31], v_2= [2,30,40], v_3 = [3,40,50], v_4 = [4,50,6]. A plaintext message, for example, is the vector x = [4,5,6] of R^3. Then:

[0037] $L(x, v_1) = 295$, $L(x, v_2) = 398$, $L(x, v_3) = 512$, $L(x, v_4) = 302$.

[0038] Furthermore,

[0039] $L(v_1, v_1) = 1403$, $L(v_2, v_2) = 2504$, $L(v_3, v_3) = 4109$, $L(v_4, v_4) = 2552$.

[0040] Therefore,

[0041] $S_1(x) = [4,5,6] - 2\cdot(295/1403)\cdot[1,21,31] = [3.579472559, -3.831076265, -7.036350677]$

[0042] $S_2(x) = [4,5,6] - 2\cdot(398/2504)\cdot[2,30,40] = [3.364217252, -4.536741214, -6.715654952]$

[0043] $S_3(x) = [4,5,6] - 2\cdot(512/4109)\cdot[3,40,50] = [3.25237284, -4.968362132, -6.460452665]$

[0044] $S_4(x) = [4,5,6] - 2\cdot(302/2552)\cdot[4,50,6] = [3.053291536, -6.8338558, 4.579937304]$

[0045] The above fractional numbers are computed with the precision of nine decimal places after the dot. In this example the numbers will be rounded up to two decimal places after the dot, that is,

[0046] $S_1(x) = [3.58, -3.83, -7.04]$,

[0047] $S_2(x) = [3.36, -4.54, -6.72]$,

[0048] $S_3(x) = [3.25, -4.97, -6.46]$,

[0049] $S_4(x) = [3.05, -6.83, 4.58]$.

[0050] To implement the cryptosystem of this example, the user of the processor-based system 101, call her Alice, decides

to send a message to the user of the processor-based system 151, call him Bob. [It is assumed in this example that the processor-based systems 101 and 151 share the secret (i.e., available only to Alice and Bob) parameters v_1, v_2, v_3, v_4 and the (non-secret) standard dot-product L on V, defined as above]. Suppose that Alice [or the processor-based system 101] chooses k = 8 and a sequence P of k integers: P = (1, 2, 3, 4, 1, 2, 3, 4) as the outer component of the encryption key [the restrictions on P in this example are that $p_j \neq p_{j+1}$ for j = 1, 2, ..., k-1, and all p_j are between 1 and 4; therefore, P can be chosen essentially at random within these limits]. Thus the encryption key K = (P, Q) is created, where Q is the inner component comprised of the parameters v_1, v_2, v_3, v_4. Based on this encryption key K, the processor-based system 101 creates the encryption automorphism T_e. This T_e is an automorphism of the space V defined by the formula

[0051] $$T_e = S_1 \circ S_2 \circ S_3 \circ S_4 \circ S_1 \circ S_2 \circ S_3 \circ S_4,$$

[0052] where the reflections S_1, S_2, S_3, S_4 are as above. For example, suppose that Alice wants to send to Bob the message M = x = [4,5,6]. The processor-based system 101 encrypts this message using the constructed above encryption automorphism T_e. The processor-based systems

101 applies the encryption automorphism T_e to M and thus creates the encrypted message E given by

[0053] $E = T_e(M) = [3.435583316, -4.617835082, -6.623621852]$.

[0054] The above fractional numbers are computed with the precision of nine decimal places after the dot. In this example the numbers comprising E are rounded up to two decimal places after the dot, that is, E is replaced by E_{round}, where

[0055] $E_{round} = [3.44, -4.62, -6.62]$.

[0056] Then transceiver 108 sends the pair

[0057] $(P; E_{round}) = (1, 2, 3, 4, 1, 2, 3, 4; [3.44, -4.62, -6.62])$

[0058] to the processor-based system 151. In the next part of the example, decryption of the received message is described. In order to decrypt the received message $(P; E_{round})$, the processor-based system 151 creates the decryption key K' = (P';Q), where P' = (4,3,2,1,4,3,2,1), that is, P' is the reversed P, and Q is the inner component as above. Based on this decryption key K' the processor-based system 151 creates the decryption automorphism T_d of the vector space V given by

[0059] $T_d = S_4 \circ S_3 \circ S_2 \circ S_1 \circ S_4 \circ S_3 \circ S_2 \circ S_1$

[0060] The processor-based system 151 decrypts the received

message E_{round} by applying the automorphism T_d:

[0061] $M_{approx} = T_d(E_{round}) = [4.004794621, 5.000831229, 5.99630786]$.

[0062] The above fractional numbers are computed with the precision of nine decimal places after the dot. In this example processor-based system 151 rounds up these numbers to the closest integers, that is, it replaces M_{approx} by M_{round}, where $M_{round} = [4,5,6]$. This is the original message M. The fact that the coordinates of the decrypted message M_{approx} are sufficiently close to integers [that is, the distances between the coordinates and the closest integers are less than 0.01] indicates that there has not been any error during transmission of the message $(P; E_{round})$. Therefore, the cryptosystem of the present invention can also be used for detecting errors of transmission.

[0063] In a further embodiment of the invention the reflections S_i will be replaced by the twisted eflections T_i in order to further enhance the security level of the proposed cryptosystem. A twisted reflections embodiment of the cryptosystem hereof works in the n-dimensional vector space V over the field of real numbers and a bilinear form L on V. A vector x in V can be written as an n-tuple of real numbers:

[0064] $x = [x_1, x_2, \ldots, x_n]$.

[0065] A bilinear form can be written as

[0066] $L(x, y) = \Sigma l_{i,j} \cdot x_i \cdot y_j$,

[0067] where the summation is over all pairs (i,j) such that $1 \le i,j \le n$, and all $l_{i,j}$ are real numbers. The embodiment of the cryptosystem hereof depends on discrete parameters n and m, which are positive integers, and two sets of continuous parameters: any vectors $v_1, v_2, \ldots, v_m$ of V and polynomial or (everywhere defined) rational automorphisms $g_1, g_2, \ldots, g_m$ of V. In an embodiment the coordinates of the vectors of the cryptosystem hereof are presented by decimal real numbers having totally l decimal digits (therefore, the average number of digits in each coordinate is $l/(n \cdot m)$. Therefore, the security level of the cryptosystem hereof provided by the first set of parameters alone is measured as the number of all such sets of vectors, i.e.,

[0068] $10^l \cdot (l - 1)!/[(n \cdot m - 1)!(l - n \cdot m)!]$.

[0069] For example, if $n = 3$, $m = 4$, $l = 72$, the security level is measured as

[0070] $10^{72} \cdot (72 - 1)!/[(3 \cdot 4 - 1)!(72 - 3 \cdot 4)!] \approx 2.5 \cdot 10^{84}$.

[0071] (Actually the security level is much higher because the total number l of the digits is arbitrary big and not public.) In one embodiment when the polynomial or rational automorphisms $g_1, g_2, \ldots, g_m$ are not public, they additionally enhance the security level of the cryptosystem. In another embodiment when the polynomial or rational automorphisms $g_1, g_2, \ldots, g_m$ are public, their contribution to security consists of an additional defense against attacks on transmitted messages. More precisely, it is much harder to reconstruct the decryption automorphism T_d that is a non-linear (e.g., polynomial or rational) transformation of V than the decryption automorphism that is a linear transformation of V, i.e., an automorphism that is a matrix.

[0072] The following is an example of an embodiment in accordance with the invention of a symmetric cryptosystem. The small numbers $n = 3$, $m = 4$, $l \leq 24$ are used for ease of illustration, however, even with these small numbers the cryptosystem hereof is still cryptographically secure. The automorphisms g_1, g_2, g_3, g_4 are considered public. Thus, in this example, the security level is measured as $1.3 \cdot 10^{30} \approx 2^{100}$. In creating a symmetric cryptosystem in accordance with an embodiment hereof (and with the previously indicated small numbers for ease of illustration), a

first step is to choose integer parameters m, n. Take, for example $n = 3$, $m = 4$. Next, the bilinear form L is chosen to be the standard Euclidean dot product on $V = R^3$, that is,

[0073] $L(x, y) = x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3$

[0074] for all x and y in R^3. Some sequence of vectors v_1, v_2, v_3, v_4 is chosen as follows: v_1=[1,21,31], v_2= [2,30,40], v_3 = [3,40,50], v_4 = [4,50,6]. And some second set of continuous parameters, i.e., the set of four automorphisms g_1, g_2, g_3, g_4, is chosen as follows:

[0075] $g_1([x_1, x_2, x_3]) = [x_1, x_2, x_3]$,

[0076] $g_2([x_1, x_2, x_3]) = [x_1, x_2, x_3]$,

[0077] $g_3([x_1, x_2, x_3]) = [x_1, x_2, x_3]$,

[0078] $g_4([x_1, x_2, x_3]) = [x_1, x_2 + f(x_1), x_3]$, where

[0079] $f(x_1) = (2x_1^3 + 7x_1^2 + 3x_1 + 10)/(3x_1^2 + 5)$.

[0080] Then the twisted reflections T_1, T_2, T_3, T_4 are defined as above by:

[0081] $T_1 = g_1 \circ S_1 \circ g_1^{-1}$, $T_2 = g_2 \circ S_2 \circ g_2^{-1}$, $T_3 = g_3 \circ S_3 \circ g_3^{-1}$, $T_4 = g_4 \circ S_4 \circ g_4^{-1}$.

[0082] In this example $T_1 = S_1$, $T_2 = S_2$, $T_3 = S_3$, but $T_4 \neq S_4$. A plaintext message, for example, is the vector x = [4, 5, 6]

of the vector space R^3 . Then:

[0083] $L(x, v_1) = 295$, $L(x, v_2) = 398$, $L(x, v_3) = 512$, $L(x, v_4) = 302$.

[0084] Furthermore,

[0085] $L(v_1, v_1) = 1403$, $L(v_2, v_2) = 2504$, $L(v_3, v_3) = 4109$, $L(v_4, v_4) = 2552$.

[0086] Therefore,

[0087] $T_1(x)=S_1(x)=[4,5,6]-2\cdot(295/1403)\cdot[1,21,31]=[3.579472559,-3.831076265,-7.036350677]$

[0088] $T_2(x) =S_2(x) = [4,5,6] -2\cdot(398/2504)\cdot[2,30,40] = [3.364217252, - 4.536741214, -6.715654952]$

[0089] $T_3(x)=S_3(x) = [4,5,6] -2\cdot(512/4109)\cdot[3,40,50] = [3.25237284, - 4.968362132, - 6.460452665]$

[0090] $S_4(x) = [4,5,6]-2\cdot(302/2552)\cdot[4,50,6] = [3.053291536, - 6.8338558, 4.579937304]$

[0091] $g_4(x) = [4, 9.943396227, 6]$

[0092] $g_4^{-1}(x) = [4, 0.056603774, 6]$

[0093] $S_4(g_4^{-1}(x)) = [3.828118531, -2.091914592, 5.742177796]$

[0094] $T_4(x) = g_4(S_4(g_4^{-1}(x))) = [3.828118531, 2.733397735, 5.742177796]$

[0095] The above fractional numbers are computed with the precision of nine decimal places after the dot. In this example the numbers will be rounded up to two decimal places after the dot, that is,

[0096] $T_1(x) = S_1(x) = [3.58, -3.83, -7.04]$,

[0097] $T_2(x) = S_2(x) = [3.36, -4.54, -6.72]$,

[0098] $T_3(x) = S_3(x) = [3.25, -4.97, -6.46]$,

[0099] $S_4(x) = [3.05, -6.83, 4.58]$,

[0100] $g_4(x) = [4, 9.94, 6]$,

[0101] $g_4^{-1}(x) = [4, 0.06, 6]$,

[0102] $S_4(g_4^{-1}(x)) = [3.83, -2.09, 5.74]$,

[0103] $T_4(x) = g_4(S_4(g_4^{-1}(x))) = [3.83, 2.73, 5.74]$.

[0104] To implement the key creation of this example, the user of the processor-based system 101, call her Alice, decides to send a message to the user of the processor-based system 151, call him Bob. [It is assumed in this example that the processor-based systems 101 and 151 share the secret (i.e., available only to Alice and Bob) first set of parameters v_1, v_2, v_3, v_4, the (non-secret) standard dot product L on V, defined as above, and the (non-secret)

second set of parameters g_1, g_2, g_3, g_4.] Suppose that Alice [or the processor-based system 101] chooses k = 8 and a sequence P of k integers: P = (1, 2, 3, 4, 1, 2, 3, 4) as the outer component of the encryption key [the restrictions on P in this example are that $p_j \neq p_{j+1}$ for j = 1, 2, ..., k-1, and all p_j are between 1 and 4; therefore, P can be chosen essentially at random within these limits]. Thus the encryption key K = (P, Q) is created, where Q is the inner component comprised of the parameters v_1, v_2, v_3, v_4 and g_1, g_2, g_3, g_4 . Based on this encryption key K, the processor-based system 101 creates the encryption automorphism T_e. This T_e is an automorphism of the space V defined by the formula

[0105] $$T_e = T_1 \circ T_2 \circ T_3 \circ T_4 \circ T_1 \circ T_2 \circ T_3 \circ T_4,$$

[0106] where T_1, T_2, T_3, T_4 are twisted reflections, as defined above. For example, suppose that Alice wants to send to Bob the message M = x = [4,5,6]. The processor-based system 101 encrypts this message using the constructed above encryption automorphism T_e. The processor-based systems 101 applies T_e to M and thus creates the encrypted message E given by

[0107] $$E = T_e(M) = [4.42453245, 6.72134463, - 13.76860997].$$

[0108] The above fractional numbers are computed with the precision of eight decimal places after the dot. In this example the numbers comprising E are rounded up to two decimal places after the dot, that is, E is replaced by E_{round}, where E_{round} = [4.42, 6.72, - 13.77]. Then transceiver 108 sends the pair

[0109] $(P; E_{round})$ = (1, 2, 3, 4, 1, 2, 3, 4; [4.42, 6.72, - 13.77])

[0110] In the next part of the example, decryption of the received message is described. In order to decrypt the received message $(P; E_{round})$, the processor-based system 151 creates the decryption key K' = (P';Q), where P' = (4,3,2,1,4,3,2,1), that is, P' is the reversed P, and Q is the inner component as above. Based on this decryption key K' the processor-based system 151 creates the decryption automorphism T_d of the vector space V given by

[0111] $T_d = T_4 \circ T_3 \circ T_2 \circ T_1 \circ T_4 \circ T_3 \circ T_2 \circ T_1$.

[0112] The processor-based system 151 decrypts the received message E_{round} by applying the decryption automorphism T_d:

[0113] $M_{approx} = T_d(E_{round})$ = [3.99511743, 4.99555740, 6.00656969].

[0114] The above fractional numbers are computed with the pre-

cision of eight decimal places after the dot. In this example processor-based system 151 rounds up these numbers to the closest integers, that is, it replaces M_{approx} by the vector M_{round}, where $M_{round} = [4,5,6]$. This is the original message M. The fact that the coordinates of the decrypted message M_{approx} are sufficiently close to integers [that is, the distances between the coordinates and the closest integers are less than 0.01] indicates that there have not been any errors during transmission of the message $(P; E_{round})$. Therefore, the cryptosystem of the present invention can also be used for detecting errors of transmission.

[0115] FIG. 2 illustrates a basic procedure that can be utilized with a symmetric encryption system, and refers to routines illustrated by other referenced flow diagrams which describe features in accordance with an embodiment of the invention. The block 201 represents the generating of the outer component of the encryption key. The routine of an embodiment hereof is described in conjunction with the flow diagram of FIG. 3. In the present example, it can be assumed that this operation is performed at the processor-based system 101. The outer component information can be published. For example, "publishing" of the

outer component information can be performed by the sender of the encrypted message. In particular, the outer component information can be transmitted by the sender of the encrypted message along with the message. Typically, although not necessarily, each transmitted message has its own outer component of the key that is generated by the sender. In the present example, it is assumed that the user of the processor-based system 101 wants to send a confidential message to the user of processor-based system 151, and that the user of processor-based system 101 can generate this outer component of the key within processor-based system 101. The block 202 represents the routine that can be used by the message sender (that is, in this example, the user of processor-based system 101) to generate inner component of the encryption key and the corresponding encryption automorphism. This routine, for an embodiment of the invention, is described in conjunction with the flow diagram of FIG. 4. The block 203 represents the routine that can be used by the message sender (that is, in this example, the user of processor-based system 101) to encrypt the plaintext message using the encryption automorphism. This routine, in accordance with an embodiment of the invention,

is described in conjunction with the flow diagram of FIG. 5. The encrypted message is then transmitted over the channel 100 (FIG. 1). The block 204 represents the routine that can be used by the message recipient (that is, in this example, the user of processor-based system 151) to generate the decryption automorphism using the decryption key that, in its turn, is produced based on the outer component generated in the block 201 and the inner component generated in the block 202. The decryption automorphism generating routine, for an embodiment of the invention, is described in conjunction with the flow diagram of FIG. 6. The block 205 of FIG. 2 represents the routine for the decryption of the encrypted message to recover the plaintext message. In the present example, this function is performed by the user of the processor-based system 151, who employs the decryption automorphism generated in the block 204. The decryption routine, for an embodiment of the invention, is described in conjunction with the flow diagram of FIG. 7.

[0116] FIG. 3 represents generation of the outer component of the encryption key. First, the length k of the outer component is chosen in the block 301. Then the outer component P is generated in the block 302: P is a sequence (p_1,

$p_2, \ldots, p_k$) of length k each member p_j of which is an integer between 1 and m [where m is the size of the set of internal parameters]. P is generated at random in such a way that $p_j \neq p_{j+1}$ for j=1, 2, ... , k-1.

[0117] Referring now to FIG. 4, there is shown a flow diagram of the routine, as represented generally by the block 202 of FIG. 2, for generating the inner component of encryption key and the corresponding encryption automorphism T_e. The routine can be utilized, in the present example, for programming the processor 102 of the processor-based system 101. The block 401 represents the choosing of a positive integer n. As first described above, n determines the dimension of the vector space V over the field of real numbers. The block 402 represents the generation of L, which is the bilinear form on the n-dimensional vector space V. In the simplified example above, L was a standard Euclidean dot product on V. Next, the block 403 represents the choosing at random vectors $v_1, v_2, \ldots, v_m$. These vectors serve as internal parameters of the cryptosystem and, in this embodiment they comprise the inner component Q of the encryption key. The coordinates of the vectors may, for example, be chosen using a random number generator, which can be implemented, in known

fashion, using available hardware or software. In the present embodiment, each of the processor-based systems is provided with a random number generator, designated by the blocks 109 and 159 respectively, in FIG. 1. The block 404 represents computation of the squares of the vectors $v_1, v_2, \ldots, v_m$ with respect to the bilinear form L . If $L(v_p, v_p) = 0$ for at least one index p, the block 403 is re-entered, and a new corresponding vector v_p is chosen. The loop 405 is continued until all the squares become non-zero. [The probability of emerging a square equal 0 is extremely low. Moreover, if L is a standard Euclidean dot product, each non-zero vector of V has a positive (hence, non-zero) square with respect to the dot product and, therefore, the loop 405 does not take place.] The block 406 is then entered, this block is representing the generation of reflections $S_1, S_2, \ldots, S_m$ relative to the vectors $v_1, v_2, \ldots, v_m$ respectively according to

[0118] $$S_p(x) = x - [2L(x,v_p) / L(v_p, v_p)]\cdot v_p$$

[0119] for p=1,2, ... , *m* as first described above. The block 407 represents construction of the encryption automorphism T_e by multiplying reflections $S_1, S_2, \ldots, S_m$ in the order prescribed by the outer component $P = (p_1, p_2, \ldots, p_k)$, in accordance with

[0120] $T_e = S_{p1} \circ S_{p2} \circ \ldots \circ S_{pk}$

[0121] as first described above [that is, T_e is obtained by multiplying the reflections $S_1, S_2, \ldots, S_m$ in the order prescribed by the outer component $P = (p_1, p_2, \ldots, p_k)$.]

[0122] FIG. 5 is a flow diagram, represented generally by the block 203 of FIG. 2, of a routine for programming a processor, such as the processor 102 of the processor-based system 101 (FIG. 1) to implement encryption of a plaintext message M. The message to be encrypted is input (block 501). The encrypted message, E, can then be computed (block 502) as $E = T_e(M)$, where T_e is the encryption automorphism constructed in the block 407 of FIG. 4. The encrypted message can be transmitted (block 503) over channel 100 to the recipient who, in the present example, is the user of the processor-based system 151.

[0123] FIG. 6 is a flow diagram of the routine, as represented generally by the block 204 of FIG. 2, for generating the decryption automorphism. The routine can be utilized, in the present example, for programming the processor 152 of the processor-based system 151. It can be assumed in the present example that, prior to receiving the message, the recipient of the message possesses the parameters of the cryptosystem: the vector space V, the bilinear form L,

and a set of internal parameters: the vectors $v_1, v_2, \ldots, v_m$ that, in the present embodiment, comprise the inner component Q. [In particular, the set of private parameters $v_1, v_2, \ldots, v_m$ can be communicated to the recipient over a secure channel of communication.] The block 601 represents inputting the parameters [that is, V, L, and $v_1, v_2, \ldots, v_m$] into the processor-based system 151. The block 602 is then entered, this block represents the generation of reflections $S_1, S_2, \ldots, S_m$ relative to the vectors $v_1, v_2, \ldots, v_m$ respectively according to

[0124] $$S_p(x) = x - [2L(x,v_p) / L(v_p, v_p)]\cdot v_p$$

[0125] for p=1, 2, ... , *m* as first described above. The block 603 represents construction of the decryption automorphism T_d by multiplying reflections $S_1, S_2, \ldots, S_m$ in the order opposite to that of the outer component $P = (p_1, p_2, \ldots, p_k)$, in accordance with

[0126] $$T_d = S_{pk} \circ \ldots \circ S_{p2} \circ S_{p1}$$

[0127] as first described above. [In other words, the construction of the decryption automorphism T_d proceeds in the same way as the construction of the encryption automorphism T_e but in the order prescribed by the sequence $P' = (p_k, p_{k-1}, \ldots, p_1)$ which is the reversed outer component $P = (p_1, p_2,$

..., p_k).]

[0128] FIG. 7 is a flow diagram, represented generally by the block 205 of FIG. 2, of a routine for programming a processor, such as the processor 152 of the processor-based system 151 (FIG. 1) to implement decryption of a received encrypted message E. The message E is received (block 701). The decrypted message M can then be computed (block 702) as $M = T_d(E)$, where T_d is the decryption automorphism constructed in the block 603 of FIG. 6.

[0129] FIG's. 8 and 9 are flow diagrams relating to the above-described twisted reflections embodiment. FIG. 8 is a flow diagram of the routine, as represented generally by the block 202 of FIG. 2, for generating the inner component of encryption key and the corresponding encryption automorphism T_e. As above, the routine can be utilized, in the present example, for programming the processor 102 of the processor-based system 101. The block 801 represents the choosing of a positive integer n. As first described above, n determines the dimension of the vector space V over the field of real numbers. The block 802 represents the generation of L, which is the bilinear form on the n-dimensional vector space V. In the simplified example above, L was a standard Euclidean dot product on

V. Next, the block 803 represents the choosing at random vectors $v_1, v_2, \ldots, v_m$. These vectors serve as the first set of the internal parameters of the cryptosystem. The coordinates of the vectors may, for example, be chosen using a random number generator, which can be implemented, in known fashion, using available hardware or software. In the present embodiment, each of the processor-based systems is provided with a random number generator, designated by the blocks 109 and 159 respectively, in FIG. 1. The block 804 represents computation of the squares of the vectors $v_1, v_2, \ldots, v_m$ with respect to the bilinear form L. If $L(v_p, v_p) = 0$ for at least one index p, the block 803 is re-entered, and a new corresponding vector v_p is chosen. The loop 805 is continued until all the squares become non-zero. [The probability of emerging a square equal 0 is extremely low. Moreover, if L is a standard Euclidean dot product, each non-zero vector of V has a positive (hence, non-zero) square with respect to the dot product and, therefore, the loop 805 does not take place.] The block 806 is then entered, this block represents the generation of reflections $S_1, S_2, \ldots, S_m$ relative to the vectors $v_1, v_2, \ldots, v_m$ respectively according to

[0130] $$S_p(x) = x - [2L(x,v_p) / L(v_p, v_p)] \cdot v_p$$

[0131] for p=1, 2, ..., m as first described above. The block 807 represents selection of a set of polynomial or rational automorphisms $g_1, g_2, \ldots, g_m$ of the vector space V. These automorphisms serve as the second set of the internal parameters of the cryptosystem. These automorphisms (along with the first set of internal parameters $v_1, v_2, \ldots, v_m$) form the inner component Q of the encryption key. The automorphisms are chosen at random as compositions of linear automorphisms of V and the basic polynomial automorphisms of the form described above:

[0132] $g(x_1, x_2, \ldots, x_n) = (x_1, x_2 + f_1(x_1), x_3 + f_2(x_1, x_2), \ldots, x_n + f_{n-1}(x_1, x_2, \ldots, x_{n-1}))$,

[0133] where $f_j : R^j \rightarrow R$ for j = 1, 2, ..., n−1 are rational maps. Each of the maps f_j is chosen recursively at random using, for example, a random number generator, which can be implemented, in known fashion, using available hardware or software. In the present embodiment, each of the processor-based systems is provided with a random number generator, designated by the blocks 109 and 159 respectively, in FIG. 1. The block 808 represents generation of the twisted reflections $T_1, T_2, \ldots, T_m$ in accordance with $T_p = g_p \circ S_p \circ g_p^{-1}$ for p = 1, 2, ..., m. The block 809 represents construction of the encryption automorphism T_e in

accordance with

[0134] $T_e = T_{p1} \circ T_{p2} \circ \ldots \circ T_{pk}$

[0135] as first described above [that is, T_e is obtained by multiplying the twisted reflections T_1, T_2,...., T_m in the order prescribed by the outer component $P = (p_1, p_2, \ldots, p_k)$.]

[0136] FIG. 9 is a flow diagram of the routine, as represented generally by the block 204 of FIG. 2, for generating the decryption automorphism T_d of the present twisted reflections embodiment. The routine can be utilized, in the present example, for programming the processor 152 of the processor-based system 151. It can be assumed in the present example that, prior to receiving the message, the recipient of the message possesses the parameters of the cryptosystem: the vector space V, the bilinear form L, and two sets of internal parameters: the vectors $v_1, v_2, \ldots, v_m$ of V, and the polynomial or rational automorphisms $g_1, g_2, \ldots, g_m$ of V. These two sets of parameters, in the present embodiment, comprise the inner component Q. In one embodiment of the present example both the vectors $v_1, v_2, \ldots, v_m$ and the automorphisms $g_1, g_2, \ldots, g_m$ can be considered private parameters. In another embodiment, only the vectors $v_1, v_2, \ldots, v_m$ can be considered private, while the automorphisms $g_1, g_2, \ldots, g_m$ can be

considered public parameters. [In particular, the private parameters $v_1, v_2, \ldots, v_m$ can be communicated to the recipient over a secure channel of communication.] In another embodiment, only the automorphisms $g_1, g_2, \ldots, g_m$ can be considered private, while the vectors $v_1, v_2, \ldots, v_m$ can be considered public parameters. The block 901 represents inputting the parameters [that is, V, L, and $v_1, v_2, \ldots, v_m$; $g_1, g_2, \ldots, g_m$] into the processor-based system 151. The block 902 is then entered, this block represents the generation of reflections $S_1, S_2, \ldots, S_m$ relative to vectors $v_1, v_2, \ldots, v_m$ respectively according to

[0137] $$S_p(x) = x - [2L(x,v_p) / L(v_p, v_p)] \cdot v_p$$

[0138] for p=1, 2, ... , *m* as first described above. The block 903 represents generation of the twisted reflections $T_1, T_2, \ldots, T_m$ in accordance with $T_p = g_p \circ S_p \circ g_p^{-1}$ for p = 1, 2, ... , *m*. The block 904 represents construction of decryption automorphism T_d by multiplying the twisted reflections $T_1, T_2, \ldots, T_m$ in the order opposite to that of the outer component $P = (p_1, p_2, \ldots, p_k)$, in accordance with

[0139] $$T_d = T_{pk} \circ \ldots \circ T_{p2} \circ T_{p1}$$

[0140] which proceeds in the same way as the construction of the encryption automorphism T_e but in the order prescribed

by the sequence $P' = (p_k, p_{k-1}, \ldots, p_1)$ which is the reversed outer component $P = (p_1, p_2, \ldots, p_k)$.]

[0141] The invention has been described with reference to particular preferred embodiments, but variations within the spirit and scope of the invention will occur to those skilled in the art. For example, it will be understood that the internal parameters of the cryptosystem can be stored on any suitable media, for example a "smart card," which can be provided with a microprocessor capable of constructing encryption/decryption keys and performing encryption/decryption processes, so that encrypted messages can be communicated to and/or from the smart card.